CLAIMS

1. An apparatus for performing Single Instruction Multiple Data (SIMD) instructions using a single multiply accumulate (MAC) unit while minimizing the operational latency, said apparatus comprising:

a MAC unit, said MAC unit generates a first half of a data result and a second half of said data result;

a defer register stores said first half of a data result; and

a miscellaneous (MISC) unit, said MISC unit determines when to release said first half of a data result stored in said defer register to synchronize said first half of a data result with said second half of said data result.

2 The apparatus of claim 1, wherein said MAC unit generates said first half of a data result before said second half of said data result.

3. The apparatus of claim 2, further comprising:

a register file, wherein said register file receives said first half of said data result substantially currently with said second half of said data result.

4 The apparatus of claim 3, wherein said MISC unit generates an exception result if said MISC unit determines said first half of said data result is in error.

Cont A1

5 The apparatus of claim 4, wherein said MISC unit further determines if said first half of a data result stored in said defer register or said exception result is to be release to said register file.

6. A method for performing Single Instruction Multiple Data (SIMD) instructions using a single multiply accumulate (MAC) unit while minimizing the operational latency, comprising the steps of:

generating a first operand data result by said MAC unit;

inputting said first operand data result to a deferred register;

generating a second operand data result by said MAC unit;

generating an exception result by a MISC unit;

inputting said first operand data result and said second operand data result into a buffer if said MISC logic determines that said first operand data result and said second operand data result are valid; and

inputting said exception result into said buffer if said MISC unit determines that said first operand data result and said second operand data result are invalid.

7. The method of claim 6, further comprising the steps of:

latching a first operand data into said MAC unit; and

latching a second operand data into said MAC unit.

8. The method of claim 7, further comprising the steps of:

generating said first operand data result from said first operand; and

Cont A1

generating said second operand data result from a second operand data.

9. The method of claim 6, further comprising the step of::

latching a first operand and a second operand data into said MISC unit.

10. The method of claim 9, further comprising the step of::

generating said exception result from said first operand and said second operand data.

11. An apparatus for performing Single Instruction Multiple Data (SIMD) instructions using a single multiply accumulate (MAC) unit while minimizing the operational latency, said apparatus comprising:

means for generating a first operand data result;

means for inputting said first operand data result to a deferred register;

means for generating a second operand data result;

means for generating an exception result;

means for inputting said first operand data result from said deferred register and said second operand data result into a buffer if said exception result generating means determines that said first operand data result and said second operand data result are valid; and

means for inputting said exception result into said buffer if said exception result generating means determines that said first operand data result and said second operand data result are invalid.

12. The apparatus of claim 11, further comprising:

means for latching a first operand data into said first operand data result generating means; and

means for latching a second operand data into said second operand data result generating means.

13. The apparatus of claim 12, further comprising:

means for generating said first operand data result from said first operand; and

means for generating said second operand data result from a second operand data.

14. The apparatus of claim 11, further comprising:

means for latching a first operand and a second operand data into said exception result generating means.

15. The apparatus of claim 14, wherein said exception result generating means further comprises:

means for generating said exception result from said first operand and said second operand data.